`LexSearch`: An Approach for Leveraging Semantic Annotations to Searching Ontologies

Jyotishman Pathak and **James D. Buntrock** and **Christopher G. Chute**
Division of Biomedical Informatics
Mayo Clinic College of Medicine
Rochester, MN 55905, USA
{pathak.jyotishman,buntrock,chute}@mayo.edu

Abstract

Recent advancements in the semantic Web has resulted in the proliferation of a number of domain-specific ontologies and vocabularies. Consequently, the ability to effectively and efficiently search ontologies has emerged as an important problem. More often the users, ranging from naive to domain-experts, who seek ontologies for their application (e.g., analysis of patient data) either use existing search engines such as Swoogle (Ding *et al.* 2004) or solicit suggestions from peers (e.g., via mailing lists). Unsurprisingly, this is a very cumbersome and labor intensive process. Towards this end, in this paper we propose `LexSearch`—a technique that facilitates data-driven (semi-) automatic ontology search. Specifically, `LexSearch` leverages semantic annotations or tags created by the users and experts to generate a triple-based data model which is applied for selecting domain-specific ontologies. The discovered ontologies are then ranked primarily based on content coverage. An important aspect of `LexSearch` is to provide an uniform ontology-language agnostic framework for finding multiple ontologies that takes into consideration various relationships between the ontological concepts.

Introduction

Recent advancements in the semantic Web and its growing popularity has resulted in the proliferation of a number of domain-specific ontologies and vocabularies. This, coupled with the dramatic development of new applications that are based on Semantic Web technologies, has propelled the need to build approaches that enable effective and efficient discovery of ontologies—a problem commonly referred to as *ontology search*. Consequently, many approaches ranging from "Google-like" search engines (e.g., Swoogle (Ding *et al.* 2004)) to more sophisticated semantics- and context-based techniques (e.g., SemSearch (Lei, Uren, & Motta 2006), ONTOSEARCH2 (Pan, Thomas, & Sleeman 2006), OntoSelect (Buitelaar, Eigner, & Declerck 2004)) have been proposed. In general, the major focus of these techniques has been to satisfy, to varying extents, "coverage of concepts" required by the user—the ontology which covers most of the terms/concepts is typically ranked higher than others. Although concept coverage is useful and important (Sabou, Lopez, & Motta 2006), there is a need to develop techniques which take into consideration the following additional requirements for searching ontologies:

- *Combination of Multiple Ontologies*: It is not surprising that due to the wide-spread and heterogeneous nature of the semantic Web, in many cases the terms and concepts desired by the user would be present in multiple ontologies as opposed to a single ontology. For example, in the medical domain, patient data contains many concepts that are present in different vocabularies such as SNOMED-CT[1] and FMA[2]. However, the current techniques for ontology search are only able to return a ranked list of ontologies (e.g., based on concept coverage) as opposed to combinations of multiple ontologies.
- *Ability to Handle Relationships*: An ontology by definition specifies a set of concepts, relationships, and other distinctions that are relevant for modeling a domain. Consequently, the ability to consider relationships between multiple concepts is vital for effectively searching and selecting ontologies. However, this requirement is not well addressed in the existing approaches for ontology search.
- *Agnostic to an Ontology Language*: Ontologies and vocabularies are published in the semantic Web using a wide range of languages such as OWL[3], OBO[4] and RRF[5]. Consequently, a search technique should be agnostic to ontology languages and discover appropriate ontologies in an uniform manner. Additionally, it should have the ability to transform an ontology developed in one language to another (Johnson *et al.* 2005).
- *Evaluation Metrics*: At present most of the techniques evaluate (and rank) ontologies based on a "Google-like PageRank" mechanism (Ding *et al.* 2004), content coverage (Buitelaar, Eigner, & Declerck 2004), and well-formedness (Alani, Brewster, & Shadbolt 2006). Further investigation is required to incorporate more comprehensive metrics that would for instance consider peer rat-

[1] http://www.snomed.org/
[2] http://sig.biostr.washington.edu
[3] http://www.w3.org/TR/owl-ref/
[4] http://obofoundry.org/
[5] http://umlsks.nlm.nih.gov/

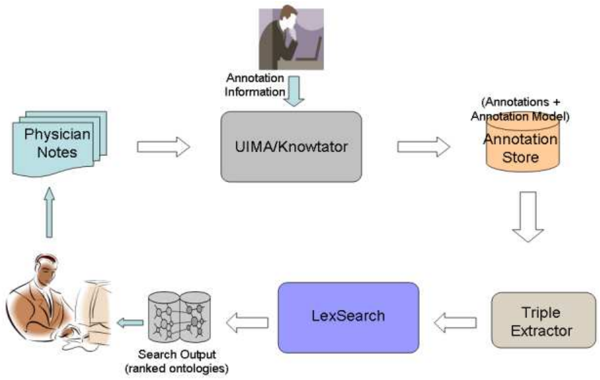

Figure 1: `LexSearch` Conceptual Diagram

ings, usability in domain-specific applications, and various non-functional aspects (e.g., is there a cost associated to use the ontology?).

- *Extraction of Ontology Modules*: In the case of large ontologies, such as the Gene Ontology[6], it is possible that the users are interested in using only a small module (e.g., molecular function) of the ontology in their application. Consequently, the ontology search techniques should have the ability to extract modules in a semantically consistent manner and return them to the user—a requirement that is severely lacking in all the current techniques for searching ontologies.
- *Performance*: As with any large scale system, performance is a bottleneck that deserves a lot of attention. Similarly, for ontology search techniques to be useful in practice, addressing issues such as scalability is of utmost importance.

Toward this end, we are developing a novel framework and a tool, called `LexSearch`, that attempts to address some of the aforementioned issues with ontology searching. In particular, in this paper we report our preliminary developments (a) in the ability to combine multiple ontologies, (b) handling of relationships between concepts during ontology search, and (c) illustrate how our search technique is agnostic to ontology languages generally used in the semantic Web. Essentially, `LexSearch` is motivated by the *data driven ontology evaluation* methodology (Brewster *et al.* 2004) which proposes that ontologies should be evaluated based on a domain-specific corpus such as a collection of a representative set of text documents. Our approach, in general, can be stated as follows (Figure 1):

- `LexSearch` assumes the existence of a corpus of text documents which are representative of a particular domain of discourse.
- These documents are first annotated (or tagged) by a domain expert based on a particular *annotation model* using existing tools such as Knowtator (Ogren 2006). The annotation model (see for example Figure 2) captures the concepts and relationships between them that are of interest.
- The generated annotations are automatically extracted and represented into a *triple-based form* (i.e., subject, property, object).
- The set of triples obtained is used for searching and ranking (at this time, primarily based on coverage) the ontologies in an *ontology-language agnostic manner*. The search mechanism takes into consideration combining *multiple ontologies* as well as *relationships* between the concepts (and their instantiations).

In what follows, we discuss the above aspects in details beginning with a brief survey of a representative set of work in the area of ontology search.

Related Work

With ever increasing number of ontologies populating the semantic Web, the ability to search ontologies effectively and efficiently has emerged as an important research topic. In particular, many efforts are directed towards developing semantic Web search engines, akin to traditional Web search engines, that allow users to search for ontologies based on keywords (e.g., concept names), examples of which include Swoogle (Ding *et al.* 2004) and OntoKhoj (Patel *et al.* 2003). These techniques typically "crawl" the semantic Web, index the available ontologies, and rank them based on various metrics including popularity (i.e., an ontology which is widely used and imported by various other ontologies is more popular compared to others).

The other spectrum of work in ontology search has adopted statistical-based approaches motivated by natural language processing (NLP) techniques. Brewster et al. (Brewster *et al.* 2004) identified important terms from a text corpus, extended them using WordNet[7], and developed a probabilistic approach to compare the ontology labels with the extended set of query terms. A similar query expansion technique using WordNet was applied in (Sabou, Lopez, & Motta 2006) to search for ontologies. However, here the query terms were represented in a "triple-like form" and those ontologies that contained the triple under consideration were returned. An important aspect of this work was to have the ability to do concept analysis based on semantic and sense disambiguation. Alani et al. (Alani *et al.* 2007) proposed the idea of leveraging Wikipedia for identifying important concepts that represent a particular domain. Thus, given a particular domain name (e.g., anatomy), their technique will first identify important terms that are relevant to the domain (e.g., hands, brain, bones) and apply these terms for searching ontologies. The authors demonstrate superior results using Wikipedia for identifying the important terms compared to using a typical Web search.

Another set of work which is very closely related to ontology search is in the area of (semantic) query answering. PowerAqua (Lopez, Motta, & Uren 2006) allows users to ask questions using (controlled) natural language constructs, which are then used to derive a set of triples for finding

[6] http://www.geneontology.org/

[7] `http://wordnet.princeton.edu/`

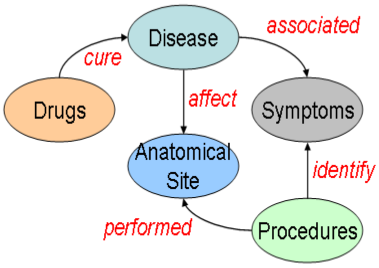

Figure 2: Sample Annotation Model

ontologies. In the event that the elements of the triple are discovered in distinct ontologies, the triple is first broken down in two or more specific triples, and then the search is re-iterated. A similar approach is proposed in the PANTO framework (Wang *et al.* 2007) where the user questions are converted into SPARQL[8] queries for querying ontologies.

As mentioned earlier, `LexSearch` is motivated by techniques based on data-driven ontology evaluation (Sabou, Lopez, & Motta 2006; Brewster *et al.* 2004; Alani *et al.* 2007). However, instead of applying purely statistical and NLP methods for analyzing the corpus, an important distinction of `LexSearch` is to leverage the annotations/tags created by humans to extract triples corresponding to the set of concepts and relationships that are representative of a particular domain of interest. These triples are then used to identify a single ontology or a combination of multiple ontologies based on semantic analysis techniques. In addition, `LexSearch` is agnostic to languages used to describe the ontologies. We explain the details of our approach in the next section.

The `LexSearch` Framework

The `LexSearch` approach, as illustrated in Figure 1, comprises of three main steps: (i) annotating the text corpus, (ii) searching ontologies based on the annotations, and (iii) ranking the searched ontologies. We discuss each of these steps in the following. In addition, we briefly discuss how `LexSearch` provides an uniform framework for searching ontologies agnostic to the language used to specify them.

Corpus Annotation

The ability to annotate a text corpora in various application domains is a well-researched topic. These approaches focus mainly on developing intelligent annotators that can (semi-) automatically process natural language constructs for named-entity recognition, co-reference resolution, template element production, and so on. However, in general, the annotators are domain dependent, and hence based on a domain-specific vocabulary/ontology (or a set thereof). For example, an annotator that can identify all the cities (mentioned in the text documents such as newspapers) in the US, may not necessarily perform well in identifying Asian cities (assuming that the vocabulary used to build the annotator is for US cities). Consequently, the ability to select a vocabulary/ontology that is representative of a particular application/domain is a significant challenge, which is becoming even more formidable with the expanding semantic Web.

[8] `http://www.w3.org/TR/rdf-sparql-query/`

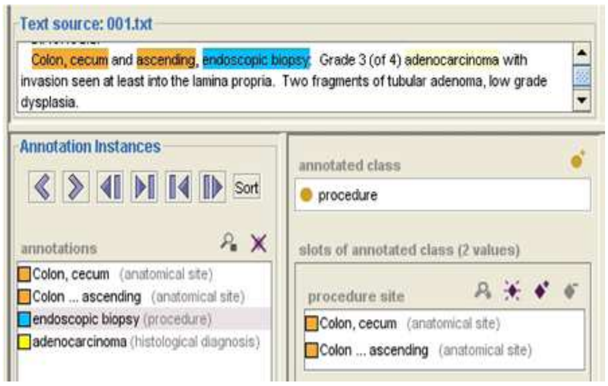

Figure 3: Knowtator screenshot of sample Annotations capturing mentions of diagnostic procedure for colorectal cancer (Ogren 2006)

The `LexSearch` approach is aimed at assisting annotator developers in finding appropriate ontologies and vocabularies. In particular, we assume the existence of an *annotation model* developed by a domain expert which captures the key concepts (and relationships between them) that are relevant for searching the ontologies. For example, Figure 2 shows a simple annotation model that captures relationships between some of the key concepts relevant in a clinical domain. The model says that a `disease` is typically *associated* with particular `symptoms` and *affect* the `anatomical sites`. The disease symptoms can be *identified* by *performing* medical `procedures` on the anatomical sites, and and in general, the disease can be *cured* by the consumption of prescribed `drugs`. Such annotation models can be built using existing tools such as Knowtator (Ogren 2006) and can potentially capture very complex relationships between the concepts. Once the annotation model is created, it can be used to annotate text documents based on human-driven analysis (Figure 3). The important advantage of this manual step is that it not only assists in capturing various relationships between the instances of the concepts, which otherwise are non-trivial to determine, but also acts as a significant step in developing annotators for automatic annotation generation (i.e., once the manually created annotations are used to search ontologies, the discovered ontologies can be applied for creating domain-specific annotators). For example, Figure 3 shows how the annotation model shown above can be applied to capture mentions of diagnostic procedures for colorectal cancer, which includes cancerous growth in the colon, cecum and apendix, that is, the procedure `endoscopic biopsy` is *performed* at anatomical sites `colon cecum` and `colon ascending`.

Annotation-based Ontology Searching

Most of the existing techniques for ontology search that leverage a text corpora (Sabou, Lopez, & Motta 2006; Brewster *et al.* 2004; Alani *et al.* 2007), apply statistical methods (e.g., TF-IDF algorithm (Salton & McGill 1986)) for identifying the query terms or concepts. Although useful in practice, these techniques fail to capture the relationships between the concepts which are vital elements in any domain model. Furthermore, depending on the technique used, it may fail to identify important concepts and/or identify irrelevant ones. For example, assume that TF-IDF is used to determine a set of diseases from a clinical text corpora comprising of 1000 relevant documents. If AIDS is mentioned only twice in the set of documents present in the corpora, it is highly likely that AIDS will *not* be identified as a disease (e.g., due to a low TF-IDF score). On the other hand, using our approach not only we are able to capture the relationships between the concepts (based on the annotation model), but also their instantiations (based on the annotations).

In particular, once the annotations are created in the corpora, they are analyzed to extract a set of *complete* and *incomplete* triples—a complete triple comprises of a subject, property, and an object; whereas incomplete triples are subject-property or property-object pairs (i.e., sets of instances that cannot be associated to any another instance in the corpora). For example, based on Figure 3, we can have two complete triples, namely, {*endoscopic biopsy, performed, colon cecum*} and {*endoscopic biopsy, performed, colon ascending*}. After the triples are extracted, the ontology search algorithm proceeds as follows:

Identifying Synonyms: The first step is to identify synonyms for all the elements present in the triples (both complete and incomplete) using WordNet. For example, the element AIDS present in a particular triple (as a subject or object) is associated with its synonym Acquired Immune Deficiency Syndrome. However, such an expansion is not very straightforward when considering the properties since they can be vaguely defined and have multiple representations (i.e., compound relations). For example, a triple "`x` *worksFor* `y`" can also be represented as "`x` *isEmployedBy* `y`" and "`x` *isAnEmployeeOf* `y`". Since in our approach, we capture the relationships based on the annotation model, a naive solution would be to assume that it is the responsibility of the domain expert (developing the annotation model) to provide the multiple representations in which the relationships between concepts are specified. However, this could be a huge overhead on the expert depending on the complexity of the annotation model. Consequently, to address this requirement, we adopt the following:

- We restrict the annotation model to comprise of "simple relations", i.e., relations comprising of multiple terms are not allowed (e.g., the relation *isEmployedBy* is not allowed).
- For each simple relation present in the annotation model, we refer to WordNet to identify synonyms and order them based on the *sense numbers*[9].

Note that the above assumptions may be overly restrictive in certain settings and we plan to leverage word sense disambiguation techniques (Agirre & Edmonds 2006) for dealing with compound relations in the future.

[9] WordNet sense numbers are assigned by frequency, so sense 1 of a word is more common than sense 2.

Performing Concept Matching: Once the set of synonyms are identified for the triples, the next step is to select ontologies that contain triples. We first consider the subjects/objects of the triples based on a *syntactic-level* matching followed by a *semantic-level* matching as described below:

During the syntactic matching process, we take into consideration the subject/object of the triples, and do a lexical analysis with the *labels* of the ontology concepts after performing standard word normalization (e.g., removing stopwords/punctuations, expanding abbreviations) (Airio 2006). For example, a subject/object `biopsy` will be matched with the concept `C15189` in UMLS since its label is `Biopsy`. In the event a perfect matching is not possible, approximate matching is done, where for example, `biopsy` might be matched with the concept `C51692` in UMLS having a label `Skin Biopsy`.[10] Note that such a simple matching process will fail in comparing a complex subject/object (such as `MRI of head with and without gadolinium`) to ontology concept labels, and hence will have to be augmented with domain-specific concept label splitting techniques—an area that we are currently investigating.

Once the syntactic matching is done, the next step tries to ensure that the matched concepts are semantically coherent. This step is based on a sense disambiguation technique proposed in (Giunchiglia, Shvaiko, & Yatskevich 2005). The main idea of our approach is to consider the synonyms (or *senses*) associated with the subjects/objects of the triples and determine if a similar sense (based on equivalence, hypernym, holonym etc.) is present in the corresponding ontology concept under consideration. In general, such senses of the ontology concepts can be determined from the ontology itself (e.g., all UMLS concepts have synonyms associated with them) or by leveraging WordNet. Note that on one hand, we reject those correspondences where a particular subject/object matches to an ontology concept only syntactically and not semantically; and on the other hand, we consider those correspondences wherein a subject/object does not match syntactically to any concept in an ontology, but share the same senses—thus, the strongest connection is when there is a syntactic and semantic correspondence.

Performing Relationship Matching: After a correspondence has been established between the subject/object of a triple, we consider the properties as follows:

In the case of complete triples, we first identify whether the triple-property matches with any of the relationships present in a particular ontology. Similar to the previous, we identify the correspondences based on both syntactic and semantic matching. Assuming that such a correspondence between a property `p` and a relationship rel has been discovered, we say that a triple `(s, p, o)` is faithfully covered in an ontology O if the following holds:

- the domain dom and the range rge of rel in O correspond

[10] Assuming that the concept `C15189` does not exist in UMLS. The technique always chooses perfect matching over approximate matching.

to the subject `s` and object `o`, respectively. That is, (`s`, `p`, `o`) corresponds to (dom, rel, rge).

Or;

- if no such (direct) correspondence is observed, then
  - there exists a concept dom' in O, where dom and dom' share the same WordNet *sense* and (dom', rel, rge) holds in O, such that `s` and `o` correspond to dom and rge, respectively; or
  - there exists a concept rge' in O, where rge and rge' share the same WordNet *sense* and (dom, rel, rge') holds in O, such that `s` and `o` correspond to dom and rge', respectively; or
  - there exists concepts dom' and rge' in O as defined above, where (dom', rel, rge') holds in O, such that `s` and `o` correspond to dom and rge, respectively.

While performing the above analysis, it might be possible that the domain dom and range rge are concepts in two different ontologies which have a relationship. In such a situation, both the ontologies will be given equal weightage (see below) in terms of covering the particular triple under consideration. Consequently, if there are many triples extracted from the annotations such that they span across multiple ontologies, then all the ontology combinations will be considered as part of the search result.

In the case of incomplete triples, the matching is performed in a similar way: when considering subject-property pairs, the objective is to find a correspondence between (`s`, `p`, `?`) with (dom, rel, rge) or (dom', rel, rge) such that dom and dom' share the same WordNet *sense*. The same process can be also applied when considering property-object pairs (i.e., (`?`, `p`, `o`)).

Note that in the above we only take into account direct and explicit relations between the concepts (either in the same ontology, or in separate ontologies). An interesting future work will be to extent this approach that can also consider implicit relationships via semantic inferencing (Rocha, Schwabe, & Aragao 2004).

Ontology Ranking

Once all the triples have been analyzed and the correspondences (if any) have been identified, the next step is to evaluate the ontologies for search result. At present, we primarily focus on ranking the ontologies based on content coverage, although we are investigating further refinement of the results based on additional constraints such as well-formedness and modularity (an ontology with a good separation of modules would be ranked higher), popularity (an ontology that is widely used and imported by other ontologies would be ranked higher), and other factors (Alani, Brewster, & Shadbolt 2006). In particular, we adopt the following process: each element in the triple carries "1 point". Thus, an ontology which covers all the elements of the triple, is assigned "3 points". The assignment of points is iterated for all the triples (both complete and incomplete) that are extracted from the annotation store. However, in the event, that a particular triple is covered by two different ontologies (i.e, the subject `s` corresponds to a concept dom in an ontology O and the object `p` corresponds to a concept rge in an ontology O' and they are related by the property `p`), then both the ontologies are assigned "2 points" each. Additionally, these ontologies are grouped together representing the fact that the the triple (`s`, `p`, `o`) can be only covered by combination of O and O'. Thus, if there are three ontologies in a repository R = $\{O, O', O''\}$, such that O scores the highest (in terms of content coverage), then assuming that there exists triples which span across O and O', the technique will return both O and O' as the combined ontology search result. Even though this represents a very primitive form of combining ontologies during search, we believe that it is an important requirement for ontology search techniques and paves the way for doing further research in this direction (e.g., extracting only the relevant modules from O and O' (Grau *et al.* 2007), and providing the combined modules as the search result).

Language-Independent Ontology Search

An important requirement for any approach to ontology search is to provide the ability to discover ontologies irrespective of the language (e.g., OWL, OBO, RRF) used to represent them. To address this end, `LexSearch` is based on the `LexGrid` (Johnson *et al.* 2005) open source vocabulary model which provides the capability to maintain the semantics of multiple ontologies and vocabulary resources in one meta-model. The current `LexGrid` system provides the ability to import ontologies in different languages and store them persistently using a standard storage format, and introduces a set of APIs that can be invoked for querying the knowledge base. Consequently, once `LexGrid` is populated with ontologies (via importing), the `LexSearch` algorithms can operate seamlessly to find relevant ontologies irrespective of their representation format.

Summary and Discussion

With the ever increasing number of ontologies and vocabularies in the Semantic Web, the ability to effectively and efficiently search them is becoming an emerging problem. To address this need, we are developing `LexSearch`—a data-driven (semi-) automatic approach for ontology search. The technique leverages a human-annotated text corpus to extract a set of triples which are then syntactically and semantically analyzed for coverage within an ontology. An important aspect of `LexSearch` is the ability to not only consider concepts and their instantiations, but also the relationships between them for finding the ontologies. Unlike the existing approaches, the technique provides combinations of multiple ontologies as search result, instead of ranking an individual ontologies (based on content coverage). Additionally, being based on an open source vocabulary model, `LexSearch` is agnostic to the languages used for ontology representation. Thus, the main contributions of `LexSearch` can be summarized as follows:

- It is an approach for searching ontologies that considers combining multiple ontologies as well as relationships between concepts (and their instantiations).

- It provides a technique for leveraging annotations or tags in documents, pertaining to relevant concepts, and applying them for searching ontologies.
- It provides an uniform framework for finding and analyzing ontologies and vocabularies independent of the language used to describe them.

At present, `LexSearch` is under active development, and as stated in the introduction, the current status of `LexSearch` covers only a partial list of the requirements for developing ontology search techniques. Even in this partial list, there are many open issues that require further investigation. For instance, currently `LexSearch` only considers direct relations between the concepts. An interesting extension in this context would be the ability to consider indirect and implicit relationships via semantic inferencing (Rocha, Schwabe, & Aragao 2004). An addendum to this is to provide the ability to handle compound relations (e.g., identify that "`x` *worksFor* `y`" also refers to "`x` *isEmployedBy* `y`") by leveraging word sense disambiguation techniques. Another topic which is of utmost importance is to build metrics for evaluating the ontology search results, which at present, are primarily based on coverage. In this context, the AktivRank algorithm (Alani, Brewster, & Shadbolt 2006) presents interesting results, and we plan to extend such an approach by other aspects of an ontology such as structural well-formedness and modularity. Orthogonal to above, another interesting area that we plan to investigate is semi-automatic generation of the annotation model. An important aspect of this work will be the extraction of relations between semantic entities from the corpus (Nguyen, Matsuo, & Ishizuka 2007).

Acknowledgment. The authors would like to thank Karin Schuler, James Masanz and Guergana Savova for participating in various discussions and providing inputs.

References

Agirre, E., and Edmonds, P. 2006. *Word Sense Disambiguation: Algorithms and Applications (Text, Speech and Language Technology)*. Springer.

Airio, E. 2006. Word Normalization and Decompounding in Mono- and Bilingual IR. *Information Retrieval* 9(3):249–271.

Alani, H.; Noy, N.; Shah, N.; Shadbolt, N.; and Musen, M. 2007. Searching Ontologies Based on Content: Experiments in the Biomedical Domain. In *4th International Conference on Knowledge Capture*, 55–62. ACM Press.

Alani, H.; Brewster, C.; and Shadbolt, N. 2006. Ranking Ontologies with AKTiveRank. In *5th International Semantic Web Conference*, 1–15. LNCS 4273, Springer-Verlag.

Brewster, C.; Alani, H.; Dasmahapatra, S.; and Wilks, Y. 2004. Data Driven Ontology Evaluation. In *International Conference on Language Resources and Evaluation*.

Buitelaar, P.; Eigner, T.; and Declerck, T. 2004. OntoSelect: A Dynamic Ontology Library with Support for Ontology Selection. In *3rd International Semantic Web Conference, Demo Session*.

Ding, L.; Finin, T.; Joshi, A.; and et al. 2004. Swoogle: A Search and Metadata Engine for the Semantic Web. In *13th ACM Conference on Information and Knowledge Management*, 652–659. ACM Press.

Giunchiglia, F.; Shvaiko, P.; and Yatskevich, M. 2005. Semantic Schema Matching. In *3rd International Conference on Ontologies, Databases and Applications of Semantics*, 347–365. LNCS 3760, Springer-Verlag.

Grau, B. C.; Horrocks, I.; Kazakov, Y.; and Sattler, U. 2007. A Logical Framework for Modularity of Ontologies. In *20th International Joint Conference on Artificial Intelligence*, 298–303.

Johnson, T. M.; Solbrig, H. R.; Armbrust, D. C.; and Chute, C. G. 2005. LexGrid Editor: Terminology Authoring for the Lexical Grid. In *AMIA Annual Symposium*, 1173.

Lei, Y.; Uren, V. S.; and Motta, E. 2006. SemSearch: A Search Engine for the Semantic Web. In *15th International Conference on Knowledge Engineering and Knowledge Management*, 238–245. LNCS 4248, Springer-Verlag.

Lopez, V.; Motta, E.; and Uren, V. S. 2006. PowerAqua: Fishing the Semantic Web. In *3rd European Semantic Web Conference*, 393–410. LNCS 4011, Springer-Verlag.

Nguyen, D. P. T.; Matsuo, Y.; and Ishizuka, M. 2007. Relation Extraction from Wikipedia Using Subtree Mining. In *22nd AAAI Conference on Artificial Intelligence*, 1414–1420. AAAI Press.

Ogren, P. 2006. Knowtator: A Plug-In for Creating Training and Evaluation Data Sets for Biomedical Natural Language Systems. In *9th International Protege Conference*.

Pan, J.; Thomas, E.; and Sleeman, D. 2006. ONTOSEARCH2: Searching and Querying Web Ontologies. In *IADIS International Conference WWW/Internet*, 211–219.

Patel, C.; Supekar, K.; Lee, Y.; and Park, E. K. 2003. OntoKhoj: A Semantic Web portal for Ontology Searching, Ranking and Classification. In *5th ACM International Workshop on Web Information and Data Management*, 58–61. ACM Press.

Rocha, C.; Schwabe, D.; and Aragao, M. P. 2004. A Hybrid Approach for Searching in the Semantic Web. In *13th International Conference on World Wide Web*, 374–383. ACM Press.

Sabou, M.; Lopez, V.; and Motta, E. 2006. Ontology Selection for the Real Semantic Web: How to Cover the Queen's Birthday Dinner? In *15th International Conference on Knowledge Engineering and Knowledge Management*, 96–111. LNCS 4248, Springer-Verlag.

Salton, G., and McGill, M. J. 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc.

Wang, C.; Xiong, M.; Zhou, Q.; and Yu, Y. 2007. PANTO: A Portable Natural Language Interface to Ontologies. In *4th European Semantic Web Conference*, 473–487. LNCS 4519, Springer-Verlag.